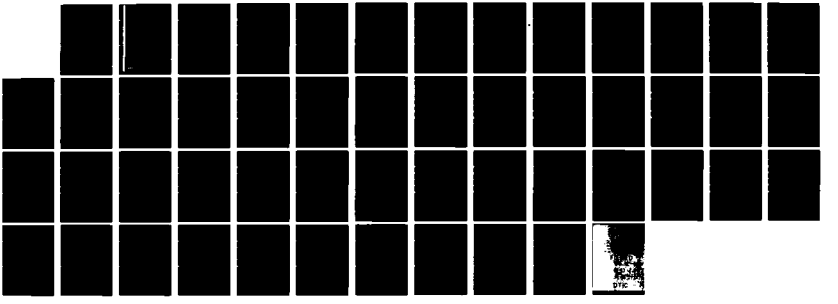
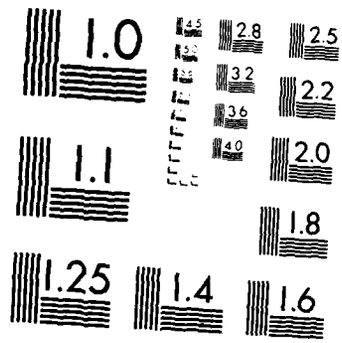
AD-A129 742 ARMY PACKET RADIO NETWORK PROTOCOL STUDY(U) SRI INTERNATIONAL MENLO PARK CA D E RUBIN NOV 77 SRI-TR-2325-143-1 DAHC15-73-C-0187 1/1

UNCLASSIFIED F/G 17/2.1 NL





ARMY PACKET RADIO NETWORK PROTOCOL STUDY

Technical Report 2325-143-1

November 1977

By: Darryl E. Rubin

Prepared for:

U.S. Army Electronics Command
Fort Monmouth, New Jersey 07703

Attn: Mr. Charles Graff, DRDCO-COM-RF-4

Contract DAHC 15-73-C-0187

SRI Project 2325

SRI International

333 Ravenswood Ave. • Menlo Park, California 94025
(415) 326-6200 • Cable: STANRES, Menlo Park • TWX: 910-373-1246

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: 2325-143-1	2. GOVT ACCESSION NO.: AD A104 742
3. RECIPIENT'S CATALOG NUMBER:	
4. TITLE (and Subtitle): Army Packet Radio Network Protocol Study	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER:
7. AUTHOR(s): Darryl E. Rubin	8. CONTRACT OR GRANT NUMBER(s): DAHC15-73C-0187
9. PERFORMING ORGANIZATION NAME AND ADDRESS: SRI International, 333 Ravenswood Avenue, Menlo Park, California 94025	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Program Code No. P62708E
11. CONTROLLING OFFICE NAME AND ADDRESS: U. S. Army Electronics Command, Fort Monmouth, New Jersey 07703, Attn: Mr. Charles Graff, DRDCO-COM-RF-4	12. REPORT DATE: November 1977 / 13. NO. OF PAGES:
	15. SECURITY CLASS. (of this report): UNCLASSIFIED
14. MONITORING AGENCY NAME & ADDRESS (if diff. from Controlling Office):	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE:

16. DISTRIBUTION STATEMENT (of this report)

[illegible]

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Packet radio network, TIDS, TACFIRE, experimental Bay Area PRNET, TCP, communication protocols, Army Message Protocol (AMP), host interface unit.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents the results of an Army Packet Radio Network (PRNET) Communication Protocol Study. Tactical Information Distribution System (TIDS) data communication requirements are analyzed and discussed, and a set of PRNET communication protocols that satisfies those requirements is defined. A general strategy for attaching a PRNET to TACFIRE is presented, and an application-level protocol is specified to make the replacement of the VHF-FM communication equipment with packet radio equipment transparent to the rest of the TACFIRE system.

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

LIST OF ILLUSTRATIONS iii

PREFACE iv

I INTRODUCTION 1

II PACKET-SWITCHED COMPUTER NETWORKS 4

III THE ARPA PACKET RADIO NETWORK 8

A. The Channel Access Protocol 10

B. Station-to-PRU Protocol 14

C. Transmission Control Protocol 15

D. TCP and User Processes 15

E. TCP Mechanisms 17

IV THE PRNET IN MILITARY APPLICATIONS 21

A. Attachment to TACFIRE 21

B. Army PRNET Communication Protocols 24

V THE ARMY MESSAGE PROTOCOL 30

A. AMP Specification 30

B. AMP Process Interface 37

C. Analysis of AMP 38

VI SUMMARY 41

REFERENCES 42

ILLUSTRATIONS

1	Overview of Current TACFIRE Configuration	2
2	Typical Computer Network	5
3	Message Transmission by Packet Switching	7
4	Packet Radio Network Logical Diagram	9
5	Layering of Packet Radio Network Protocols	11
6	Format of PRNET Packet	12
7	Format of Internet Packet	18
8	Proposed TACFIRE Communication Structure	23
9	Proposed PRNET Protocol Structure for TACFIRE Application	25
10	Host Interface Unit Process Structure	27
11	Proposed Intranet Naming Convention	28
12	Format of AMP Header	32
13	Format of Association State Block	34

PREFACE

There are enormous unfulfilled needs in the military for data communication at all levels, tactical and strategic, owing largely to the proliferation of computer-based systems. These needs are becoming well recognized; for example, the Army is aware of the emerging requirements for tactical information distribution systems (TIDS) that can serve the artillery, tanks, intelligence, air defense, logistics, and other tactical components. Current tactical systems, such as the AN/VRC-12 VHF-FM radio and the AN/TRC series of multichannel voice radio, are unable function effectively in this capacity. However, with the advanced computer communication technology now available and under development, which encompasses virtually all media (land lines, satellite, ground radio), it appears that powerful and flexible TIDS solutions can be achieved. The experimental packet radio network (PRNET), which has already demonstrated a powerful internetwork capability through its interconnection with the ARPANET, employs such a technology.

The Army is currently evaluating the possibility of utilizing the Tactical Fire Direction System (TACFIRE) in a TIDS testbed. TACFIRE's communication structure suggests that it might be desirable to introduce packet radio TIDS technology to the TACFIRE system. This could be achieved by substituting a single (shared) broadband PRNET for the several (dedicated) narrowband VHF radio nets now being used, an approach applicable to any TIDS. For compatibility, such a substitution of communication media must be transparent to the existing communication protocols and software, as well as to those components not replaced by the packet radio equipment.

The purpose of this report is to recommend a communication protocol structure for use within military PRNETs that will fulfill the communication requirements of TACFIRE and other Army tactical data systems while requiring minimum modification to these systems. (While this approach cannot take full advantage of the increased power of internetwork computer communication, it is a first step in providing the Army with a modern data communication system.) The proposed protocol structure makes use of the ARPA-developed transmission control protocol (TCP) for all internet and intercommand center message exchanges, and a new Army message protocol (AMP) specifically designed for the low-level TIDS environment, for message exchange among remote terminals, and between the remote terminals and command centers. To minimize changes to the user components, an application protocol (MSGMUX) is defined to mediate the transfer of messages between the user equipment and the PRNET equipment without modifications of any sort to the messages or message-flow patterns.

This report also proposes a strategy for attachment of a PRNET to a specific user system, the Army Tactical Fire Direction System (TACFIRE), to replace the VHF FM communication equipment currently in use. This will be done by interfacing PRNET host interface units (HIUs) to the TACFIRE components at a bus level where such attachment can be made with no effect on other components, while permitting access to all TACFIRE message flow. At the TACFIRE fire direction centers (FDCs), this proposed attachment will be made at the digital data terminal (DDT) level, displacing the DDTs. At the remote terminal level, the HIU will be interfaced to the internal bus of the terminal device, and any special input/output transformations necessary for the success of such attachment will be implemented within the HIU to avoid modification to TACFIRE terminal software.

TACFIRE was chosen as a test system for the introduction of PRNET communication technology since it is the most developed computer-based military system available; it is therefore well established, and has much available documentation. All recommendations in this report regarding TACFIRE or general military communication are purely preliminary; they are intended as bases for further design and analysis pending further investigation and initial testing of military PRNET communication systems.

I INTRODUCTION

The Army tactical fire direction system (TACFIRE) [1]* is a computer-assisted command and control system for the direction of tactical fire support. Its objective is to automate the integration of factors of strategy, tactics, intelligence, communication, and logistics in order to increase fire support effectiveness, by improving accuracy, reaction time, use of target information, and allocation of fire. TACFIRE automates the functions of the battalion and division artillery fire direction centers (FDC), as well as those in the division fire support element (FSE), including fire unit and ammunition status, meteorological data, nonnuclear fire planning, target intelligence, tactical and technical fire control, survey information, preliminary target analysis, nuclear target analysis, nuclear fire planning, chemical target analysis, and fallout prediction.

Figure 1 summarizes the TACFIRE configuration and identifies the system components. TACFIRE currently uses VHF FM communication channels, with access by contention; such radio links suffer several deficiencies, especially in data transmission applications, including:

- Functional incompatibility: The VHF radios were designed for speech rather than data traffic; selected units and repetitive field adjustments are required to achieve operating characteristics suitable for data transmission.
- Low bandwidth: Currently, the maximum data transfer rate per logical communication path is 1200 baud (less with coding).
- Low reliability: Data error rates are undesirably high, requiring excessive retransmission of messages.
- Susceptibility to interference: Narrowband communication links are easily disrupted by channel contention and random background interference (e.g., multipath), as well as by deliberate jamming and spoofing.
- Susceptibility to interception: The VHF FM radio signals are easily intercepted by the enemy for purposes of locating and homing in upon TACFIRE installations.
- Low sharing capacity: Contention access limits the number of

---

* References are listed at the end of the report.

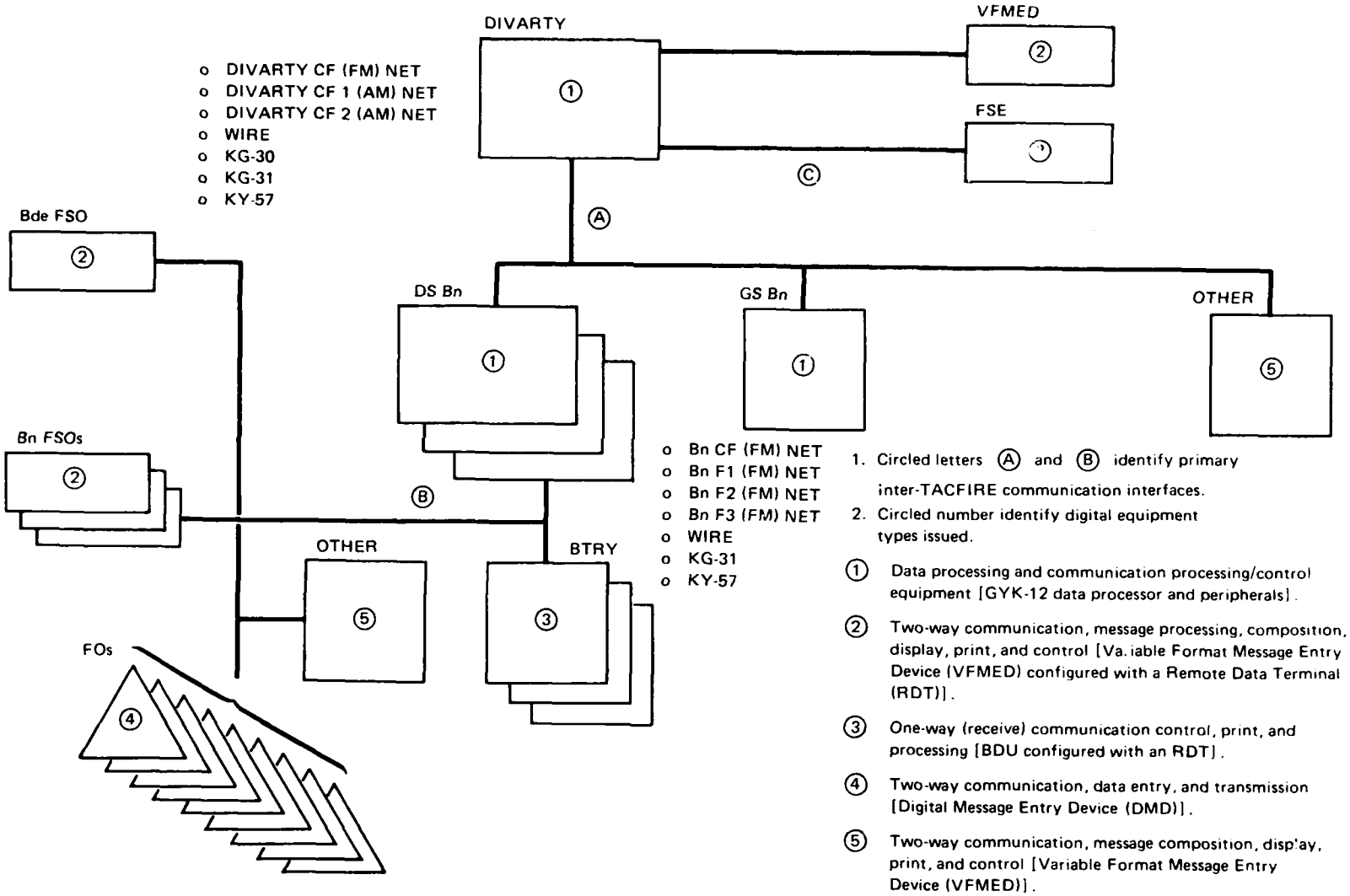


FIGURE 1 OVERVIEW OF CURRENT TACFIRE CONFIGURATION

users sharing a single radio link. This contention necessitates a large number of manually configured physical nets using separate channels, engendering communication inflexibility through difficulties in internet communication and physical net reconfiguration.

- Manual operation: The operation of the radio equipment and configuration of the communication nets must be done manually, a serious source of inefficiency and inflexibility in a computer communication environment.

TACFIRE would greatly benefit from a more sophisticated communication medium free of the above problems, and offering enhanced real-time flexibility and user capacity, as well as a capability for internet interaction with other military data systems. The state-of-the-art packet radio technology can provide this kind of performance [2].

The purpose of this report is:

(1) To show how military systems (specifically TACFIRE) could benefit from employment of packet radio technology.

(2) To recommend a set of PRNET protocols that fulfill the needs of general military data communication.

(3) To propose an approach for interfacing a PRNET to the TACFIRE system for the implementation of the Army PRNET protocols.

Information about packet switching in general and the current PRNET technology and protocols is presented as background in the following two sections.

II PACKET-SWITCHED COMPUTER NETWORKS

Computer networking technology arose from the need to share distributed computer resources, notably data bases, data storage facilities, and hardware and software computing facilities [3]. To make the resources of a set of computer systems and stand-alone terminals mutually accessible, a communication medium is required to mediate the exchange of control and data traffic between sites. Control traffic is used to request and regulate the access to remote resources, while data traffic carries the user-level information between sites for use by the acquired resources.

The communication medium provided for computer interaction must be capable of rapidly and dynamically establishing and reconfiguring logical traffic paths through the network in response to user needs or the outcome of on-going processing tasks. These paths must be high bandwidth, and provide for a reliable and efficient exchange of control and data traffic that is insensitive to local failures in the network communication equipment.

Figure 2 illustrates a typical computer network. The network consists of several computer systems, called hosts, and a set of stand-alone intelligent terminals; also shown are two teletypes interfaced to the network through a special terminal access multiplexer. These elements are all interfaced to what is known as the subnet, which serves as the common communication medium described above. Through the subnet, users at stand-alone terminals and at host terminals, as well as automated functions (processes) within the hosts themselves, can access and utilize any network host resources that are made available to the network.

The high performance requirements of the subnet normally necessitate an architecture consisting of computer-based switching nodes. Such an intelligent subnet can optimally route traffic through the network depending upon its type (critical or noncritical, control or data), automatically accounting for instantaneous global and local operational and failure conditions, and monitoring and ensuring the integrity of the transferred data.

Internode communication within the subnet may occur on multiplexed radio or landline links, but the dedication of an end-to-end (ETE) chain of link slots (a "circuit") to a single traffic path is unacceptable in computer communication applications for three reasons:

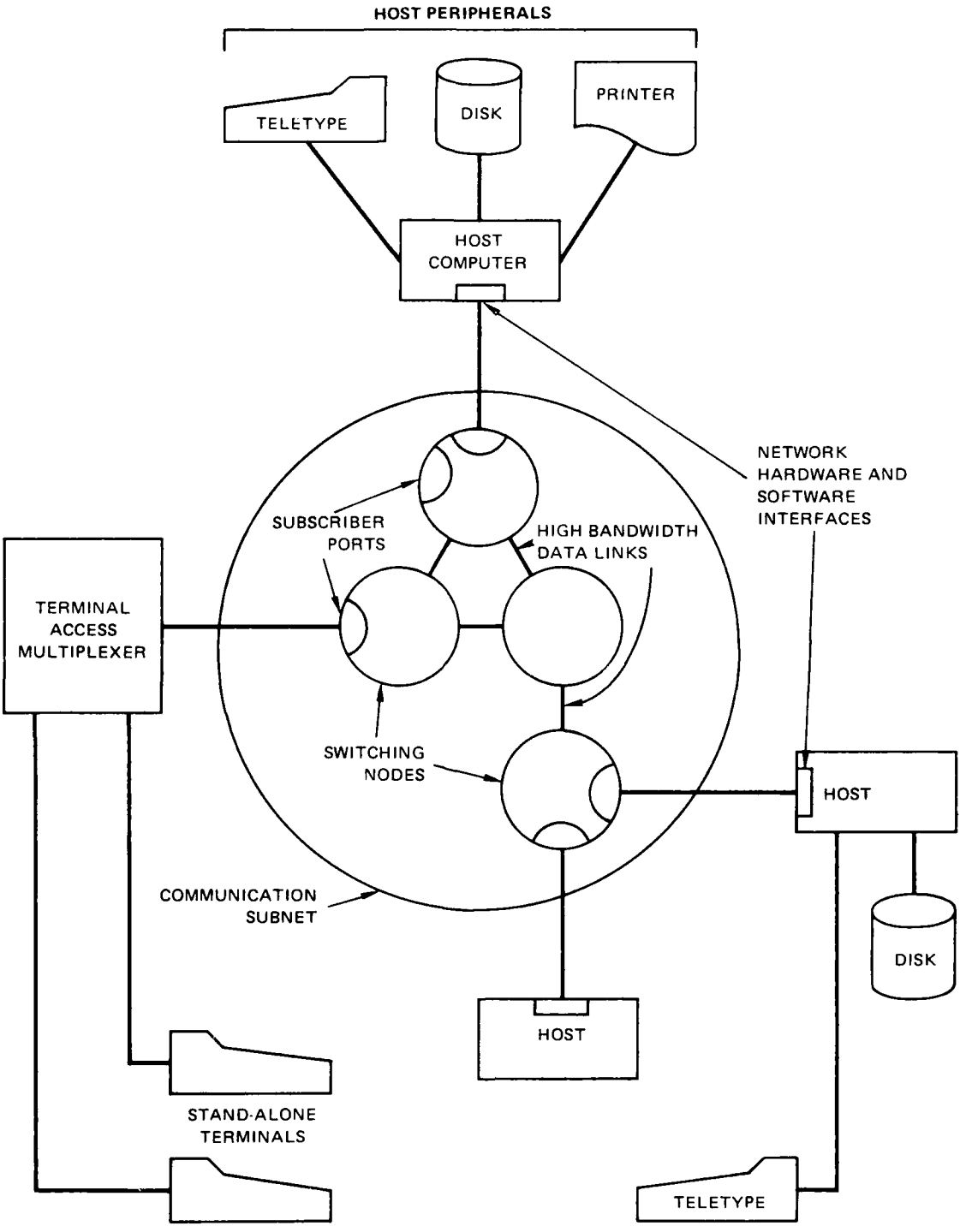


FIGURE 2 TYPICAL COMPUTER NETWORK

- The number of active logical traffic paths is potentially much larger than the number of available circuits.

- Computer communication is very bursty, so the peak bandwidth is much higher than the average bandwidth. Dedicating a circuit to a single traffic path would promote poor channel utilization.

- Effective computer interaction requires minimum ETE message delay. The time required to reserve, set up, and release circuits would degrade overall ETE performance.

In a subnet specifically designed for computer communication, slots are further multiplexed at the software level by chopping traffic streams into "packets," and switching link slot assignment among the traffic from several streams; this is known as packet switching. Division of information into packets normally occurs at the entrance to the subnet; packets on a given traffic path are individually forwarded from node to node toward their destination along the best route that avoids nonfunctional nodes, and reassembled with ETE error checking into a contiguous stream at the subnet exit (see Figure 3). To aid sequencing, routing, and control of packets within the subnet, a header is appended to each packet at the subnet entry, and removed at the subnet exit. One example of a packet-switching node that operates in this fashion is the ARPANET interface message processor (IMP) [4].

Communication protocols are needed to realize the potentials of a packet-switching subnet--to mediate traffic stream fragmentation and reassembly, routing, and ETE error checking, as well as to facilitate higher-level remote resource access functions; these protocols consist of globally understood sequences of control messages, and usually exist in a hierarchical structure. At the lowest level is the node-to-node protocol that uses the packet header information to forward a packet through the subnet from source to destination. Calling upon this level are the host-to-host and terminal-to-host protocols, which facilitate communication among the set of hosts and terminals by ensuring reliable and properly sequenced ETE information exchanges. At the highest process-to-process level, protocols directly bring about and regulate access to the remote network resources (such as files) through normal interprocess control mechanisms. Such a hierarchy of communication protocols can render the details of network operation and access fully transparent, provide for fully reliable information exchange, and facilitate simplified, flexible utilization of the distributed network resources.

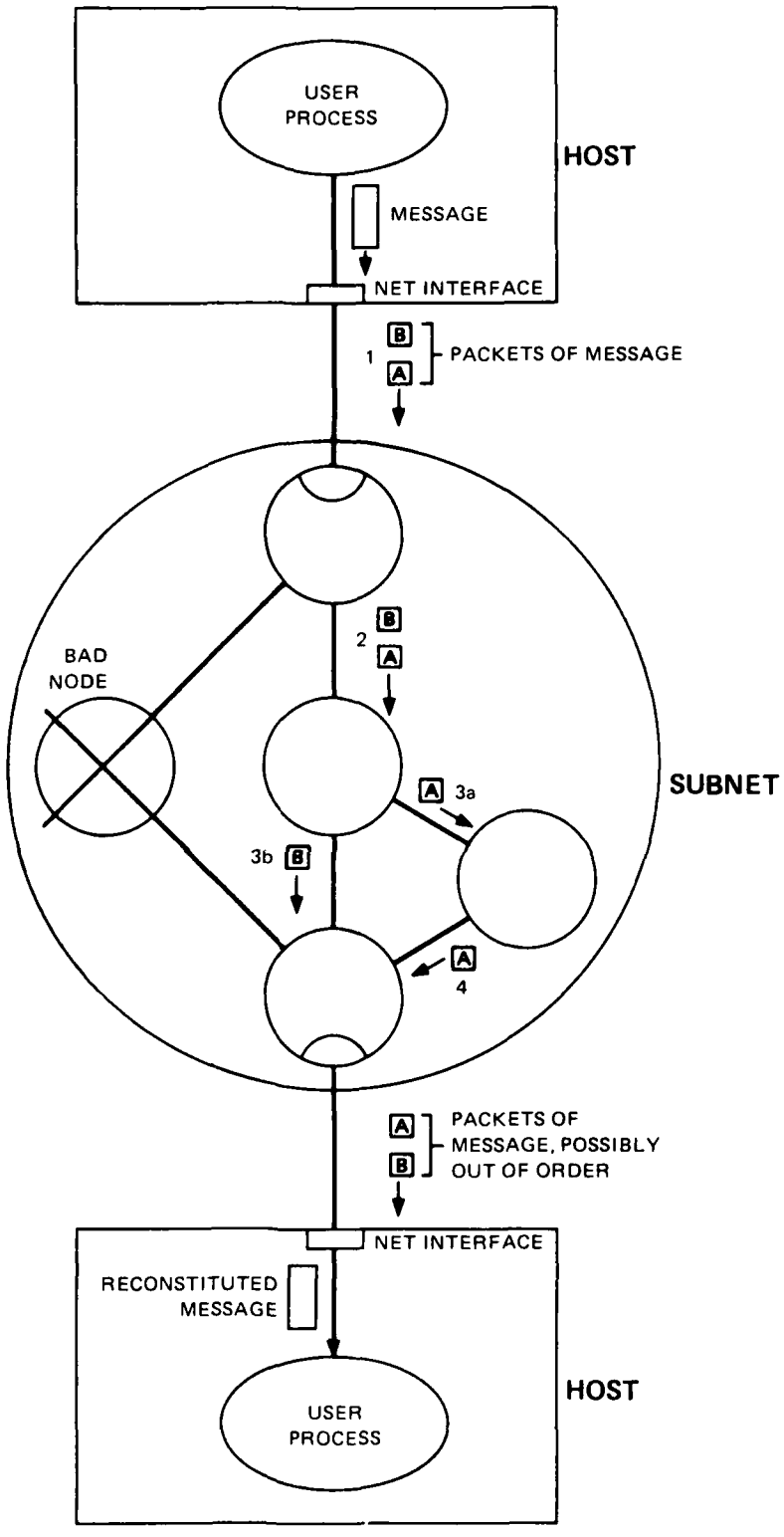


FIGURE 3 MESSAGE TRANSMISSION BY PACKET SWITCHING

III THE ARPA PACKET RADIO NETWORK

The ARPA Packet Radio Network (PRNET) combines packet-switching techniques with advances in microwave and microprocessor technology to provide a system for mobile digital communication and data distribution using radio channels. The radio technology being utilized is inherently resistant to jamming, spoofing, and detection. An experimental PRNET is currently being tested in the San Francisco Bay Area, with repeater nodes located at selected sites to provide coverage of much of the Bay Area.

A logical diagram of a PRNET is shown in Figure 4. The primary component at each subnet node in the PRNET is the packet radio unit (PRU). Each PRU contains an L-band microwave spread-spectrum radio unit for transmitting and receiving packets at either of two rates (100 and 400 kbits/s), and a microprocessor-based digital unit that implements the node-to-node protocol for controlling the routing and flow of packets between the PRUs. A PRU may operate stand-alone as a repeater, or may be attached to a computer system (host) or terminal interface unit (TIU) through a special digital (1822) interface [5]; this interface is the portal through which all packets enter and leave the network.

PRNET nodes are logically organized into a hierarchy determined by their radio "distance" (number of radio hops) from a special node, the station; the station is at Level 0, PRUs in direct contact with the station are at Level 1, PRUs that require one repeater hop to contact the station are at Level 2, and so on. The station consists of a PRU attached to a station processor (a PDP-11/40 in the SRI testbed); it is the site of centralized network control, and is responsible for monitoring of overall network connectivity, route assignment, and updating of remote PRU operating and data base parameters [6]. In a very hostile environment, multiple cooperating stations can be deployed at different sites within the network to provide fail-soft operation. The PRNET is therefore unlike other computer networks (such as the ARPANET) wherein network control functions (e.g., routing) are distributed, and accordingly is not susceptible to network-wide disruption of these functions as a result of improper node behavior.

The radio connectivity that determines the exact hierarchical structure of the PRNET at any moment is primarily a function of the geographic separation and terrain environment of the PRUs. Since PRUs can be mobile (and can fail), the logical configuration of the network can change dynamically; however, since network control is automated by the station, these topological changes are transparent to the network

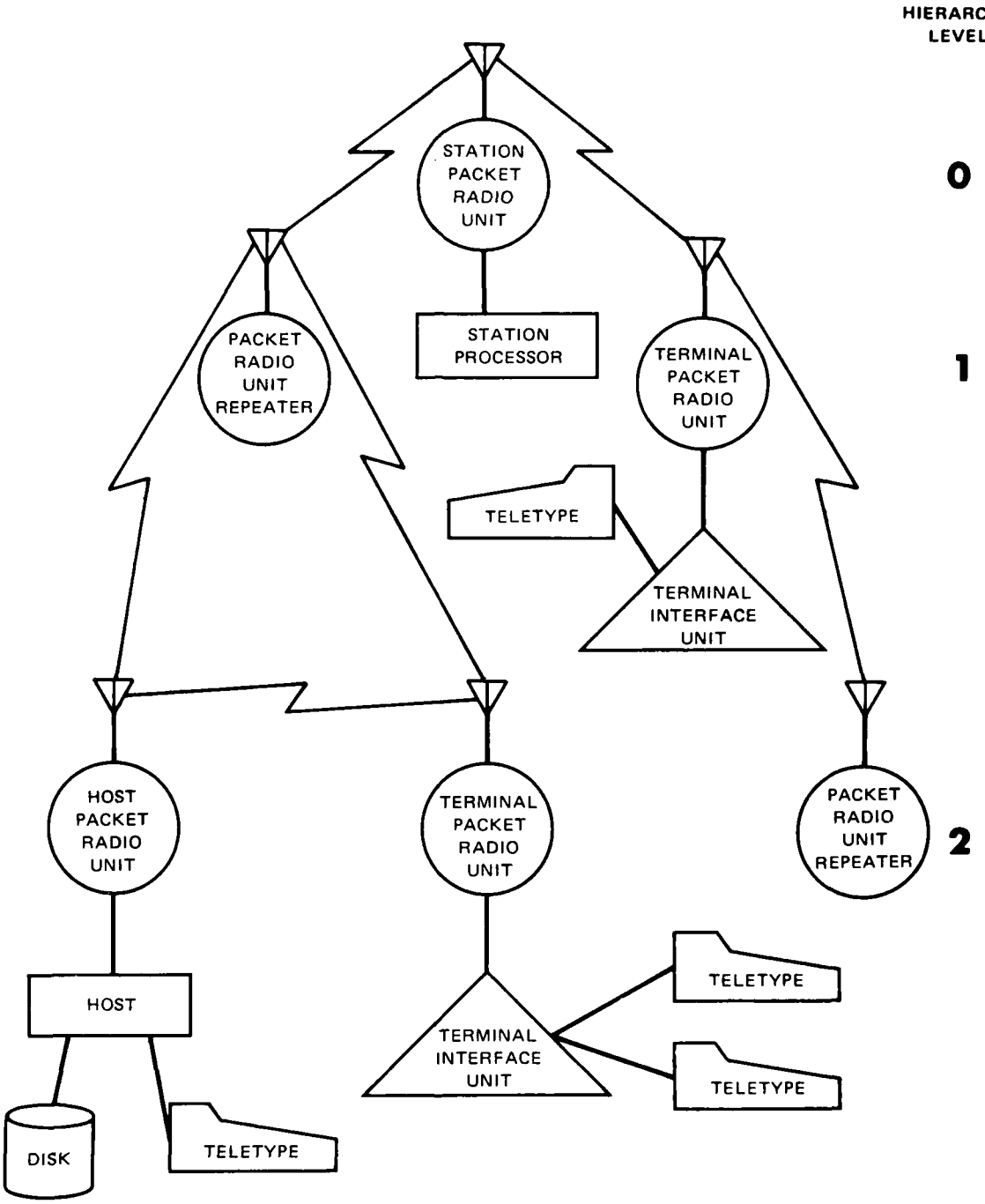


FIGURE 4 PACKET RADIO NETWORK LOGICAL DIAGRAM

users, whose processing tasks can continue uninterrupted even in the face of major network reconfiguration and distributed node failures. To ensure such robust operation is the function of the several communication protocols implemented within the PRNET nodes, station, TIUs, and hosts. These include:

- Channel Access Protocol (CAP)
- Station-to-PRU Protocol (SPP)
- Transmission Control Protocol (TCP).

This protocol layering is shown in Figure 5. The protocols are discussed in the sections that follow.

A. The Channel Access Protocol

CAP [7], which is implemented by the digital section of each PRU, provides the basic mechanism for packet transport from node to node within the PRNET. It performs functions of radio channel access, packet routing, hop-by-hop retransmission, and system monitoring. It makes use of a packet header (PRNET header, Figure 6), which is supplied by the packet source and removed at the packet destination; the header contains the addressing, sequencing, routing, and control information needed by CAP. PRNET packets are 11 to 127 16-bit words long, the first 11 of which are reserved for the PRNET header; an additional 32-bit cyclic redundancy checksum (CRC) is appended to the packet by CAP at the subnet entry for detection of radio transmission bit errors.

1. Radio Channel Access

A PRU can access the radio communication channel in either of two ways, as determined by a control message from the station. In the ALOHA mode [8], packets are transmitted spontaneously, risking packet collisions from simultaneous channel access, and an acknowledgment of correct receipt by the next node is awaited. If the acknowledgment does not arrive within a specified time interval, the packet is presumed lost, and retransmitted after a randomized time-out interval (to prevent repetitive collisions). (Of course, lost packets may actually have been discarded by the receiving node owing to bit errors, but this is of no import to the sender.) In the carrier-sense mode [9], PRUs wait until the radio channel is idle before transmitting a packet, improving channel utilization by avoiding collisions (since the packet transmission time is only a few milliseconds, PRUs normally wait no more than tens of milliseconds to transmit a packet when there is channel contention).

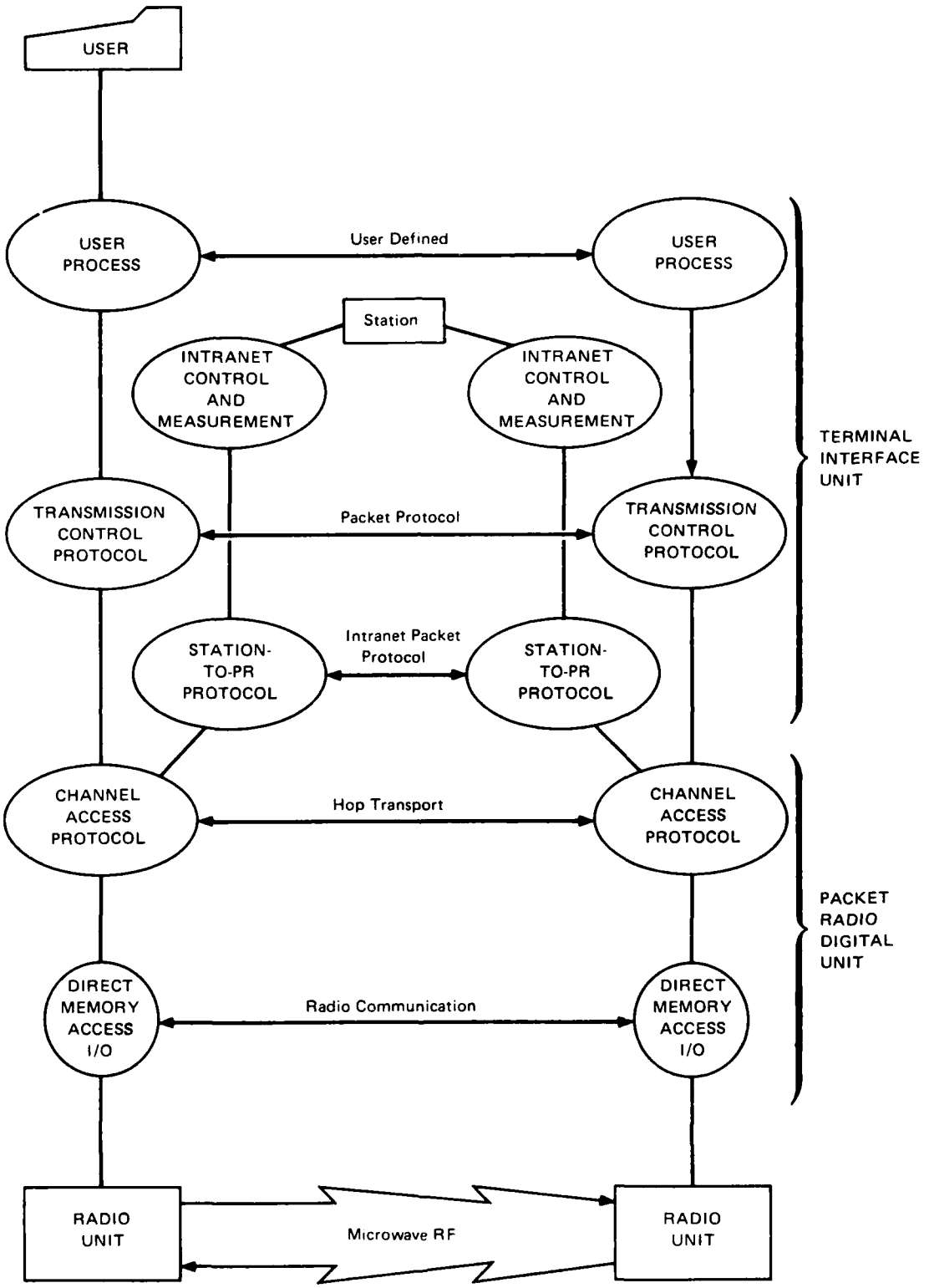


FIGURE 5 LAYERING OF PACKET RADIO NETWORK PROTOCOLS

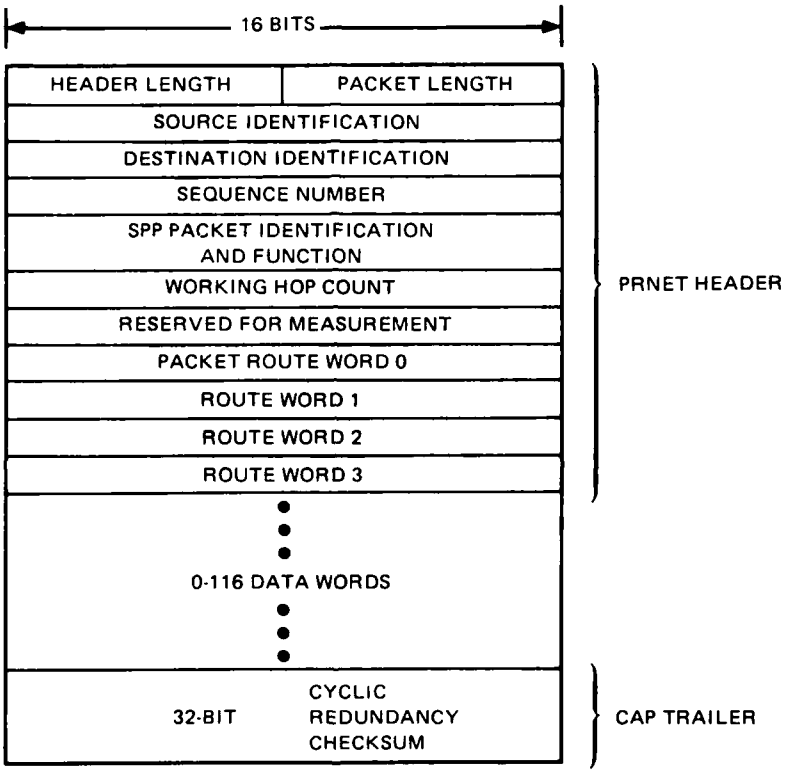


FIGURE 6 FORMAT OF PRNET PACKET

2. Routing

The PRNET currently employs a centralized routing algorithm, which requires minimal routing table space at each node. The station assigns a logical address (label) to each node, and a route along which it will forward packets to the station. Assigned routes always carry packets one hierarchy level closer to the station with each hop; when the packets arrive at the station, they are forwarded along an outbound route (chosen by the station), which carries them one level further from the station with each hop until they arrive at the destination node. The PRNET header contains space for the list of logical addresses along the attached packet's route.

3. Hop-by-Hop Retransmission

Packets may be lost as they travel between nodes; possible causes include packet collisions resulting from channel contention, sudden connectivity changes, and random fluctuations in ambient noise and propagation conditions. To improve the probability of successful multihop transport, CAP performs hop-by-hop retransmission and positive acknowledgment. When a PRU forwards a packet, it saves a copy, which it will retransmit at a later time unless it receives an acknowledgment of correct receipt of the packet; this acknowledgment is implicit if the PRU hears the packet being forwarded at the next hop, and explicit if only the packet's header is transmitted (explicit acknowledgments are used at the last hop, from which a packet is not forwarded, and at a PRU that is transmitting and receiving at different bit rates). Retransmissions can recur any number of times up to a set maximum, after which the PRU will discard the packet; because CAP (and consequently the packet radio subnet) does not guarantee packet delivery, reliability must be ensured by higher level protocols.

4. System Monitoring

Because of the dynamic variability of the PRNET's physical and logical configuration, each PRU periodically announces its presence to the station by emitting special "repeater-on" packets (ROPs). Any node that receives a ROP enters its own label into the packet header and forwards the packet toward the station. This mechanism provides the station with a record of all active PRUs at a given moment, and all the routes that lead to them. This information allows the station to maintain its network connectivity and routing tables, automatically updating them whenever existing nodes change or lose connectivity, and whenever new nodes establish connectivity.

B. Station-to-PRU Protocol

The SPP [10] is an ETE protocol designed specifically for intranet applications within a network with the operational characteristics of the PRNET. It provides for reliable packet-at-a-time communication on sequenced, full-duplex connections, but without any coupling between the two data directions, and without any traffic flow control mechanism. A full-duplex connection functionally consists of two independent simplex connections, one in each direction, identified by matching PRNET source/destination ID pairs at the communicating sites.

The SPP shares the PRNET header with CAP, a design decision which curtails packet overhead but also limits SPP's flexibility and generality. However, such power is unnecessary since the SPP's sole function is to reliably mediate control communication between the PRUs and the station. For example, all label and route information distributed by the station to the PRUs is carried by SPP connections.

To open a connection, two SPPs exchange "open" control messages for establishing mutually acceptable initial sequence numbers (ISNs) for the connection, one in each direction; successive packets sent must bear sequence numbers that monotonically increase from the ISN for that direction. When a packet is received by the destination SPP, the packet sequence number is used to determine whether the packet is the next in sequence; if so, its correct arrival is acknowledged by returning a special control message (ACK) bearing the packet's sequence number. Out of sequence packets from the past (sequence number lower than the expected sequence number) are acknowledged and discarded, while those from the future may be discarded or saved and acknowledged when the preceding packets in the sequence have arrived.

The SPP achieves communication reliability through packet retransmissions: a packet sent is saved by the source SPP until a corresponding ACK is received. If no ACK arrives within a specified time interval, the packet is retransmitted; otherwise, it is discarded. Of course, after a set number of unsuccessful retransmission attempts, the SPP will discard the undeliverable packet and communicate its failure to the using process.

The SPP utilizes other, similar time-out mechanisms to promote robust operation in the face of network and host failures; the values of the time-out intervals are highly dependent upon the operational parameters of the network, including distributions of packet lifetime, ETE delay, and node/host failure (hard outage) duration.

C. Transmission Control Protocol

The TCP [11] was created to fulfill the need for fully reliable ETE internet communication [12]; as a protocol applicable to more than one network, its operational characteristics had to be network-independent. Important TCP design considerations included:

- Standardization of the network interface presented by the TCP to insulate user processes from specificities in the packet transport facilities of the host network.
- Expanded addressing to achieve flexible and unique source/destination logical naming.
- Packet fragmentation at the juncture (gateway) between dissimilar networks, with fragment reassembly at the destination TCP.
- Functional independence from network operational characteristics.
- Robust operation despite random network or gateway failures.
- Effective ETE flow control.

D. TCP and User Processes

The TCP assumes that all ETE communication is between two processes, and that processes communicate across "ports" with logical names that may be globally known or dynamically assigned. Processes exchange "letters" of unspecified size and content over their ports, and the TCP guarantees that all letters sent will be delivered so long as there is a viable communication path between the letter source and destination.

Since process port names are selected independently by each operating system, TCP, or user, they may not be unique. To identify process ports uniquely at each TCP, an expanded port name called a "socket" is created by the TCP by concatenating a network identifier and a TCP identifier to the port name. Therefore, the full-duplex logical communication path (connection) between two process ports is uniquely specified by matching [local-socket,foreign-socket] pairs at the two TCPs serving those ports.

Connections are established when two processes wishing to communicate each issue a TCP "open," specifying the local and foreign

socket names. The two TCPs will receive the matching opens, and after several control messages have been exchanged, the connection will be ready for use. Sometimes, a process offering a globally known service may be willing to accept connection requests from any foreign process, in which case it will not be able to specify a foreign socket name in its open call; instead, it leaves the foreign socket unspecified, and the TCP fills it in as soon as a foreign connection request addressed to the "listening" socket is received.

Once a connection has been opened, processes can exchange letters; the TCP "send" call is provided for sending letters, and the "receive" call is provided for receiving them. With each call is supplied the number of 8-bit "octets" that the calling process wishes to send or receive. Since there is no limit on this number, the TCP may have to chop the letter (or fragment thereof) into pieces ("segments") that will fit within the packets of its resident (host) network. Of course, as segments cross gateways, they may be further divided (into "fragments") before injection into the next network. The letter is ultimately reconstituted at the destination TCP by reassembly of the various fragments and segments.

Connections may be disbanded by the process at either side of the connection through the TCP "close" call. A close results in the exchange of TCP control messages that release the sockets at both ends for reassignment, but not before all sent letters have been delivered and acknowledged.

Three other functions complete the TCP user interface:

(1) Interrupt: This call causes the remote process to be interrupted from the on-going exchange of letters by transmission of a special interrupt (INT) packet. (One of its uses is to simulate a "break" signal from a terminal.)

(2) Status: This call returns local status information pertaining to the connection.

(3) Abort: This call aborts a connection, regardless of its state, flushing all outstanding data.

E. TCP Mechanisms

To achieve internet addressing, connection management, sequencing, fragmentation, reassembly, flow control, and bit-error detection, an internet packet format (Figure 7) has been defined for use by the TCP; the packet consists of an internet addressing header, a TCP header, optional data segment, and checksum. Internet packets are embedded within the header and trailer formats of the local network before they are delivered to the attached packet switch as a "local packet." Local packets received by a gateway for forwarding to another network are extracted from the local packet, leaving the internet packet, which is then repackaged by the gateway as one or more (if fragmentation is necessary) local packets for the next network.

1. Traffic Management

Since it is a reliable communication medium, the TCP is capable of detecting packet loss, bit-error, duplicate, and out-of-order conditions, with a positive acknowledgment/retransmission (PAR) scheme [13] to ensure delivery. TCP's primary innovation is its use of a windowing mechanism for sequencing and flow control, wherein the control flag octet of the header and every successive octet of data has a unique, monotonically increasing sequence number;* this permits gateways to fragment packets as needed to get them across networks with small packet sizes. Acknowledgments are cumulative, so that an acknowledgment of sequence number "X" indicates that all octets up to but not including "X" have been received; duplicate detection in the presence of retransmission is therefore straightforward.

A transmission window is defined to have a lower bound and a size (the upper bound is implicitly equal to the lower bound plus the window size). The lower bound is set at connection establishment time (window synchronization), and the window size is chosen locally by each TCP, but may be negotiated by the TCPs at any time. To be accepted as in sequence, a received packet's sequence number range must lie entirely within the current receive window; if it does not, only the part overlapping the window is kept, and the rest is discarded. Duplicate octets are also discarded; octets are known to be duplicates if their sequence numbers within the window have been marked as received and not yet acknowledged.

---

* For practicality, the sequence number space is expressed as a 32-bit integer. The sequence space is therefore very large but finite, so all arithmetic dealing with sequence numbers can be performed modulo 2**32; this unsigned arithmetic preserves the ordered relationship of sequence numbers as they cycle from 2**32-1 to 0. Since the sequence space wrap-around (hours) time is much greater than the largest assumed packet lifetime (tens of minutes), it is reasonable to expect that 32-bit sequence numbers will be unique.

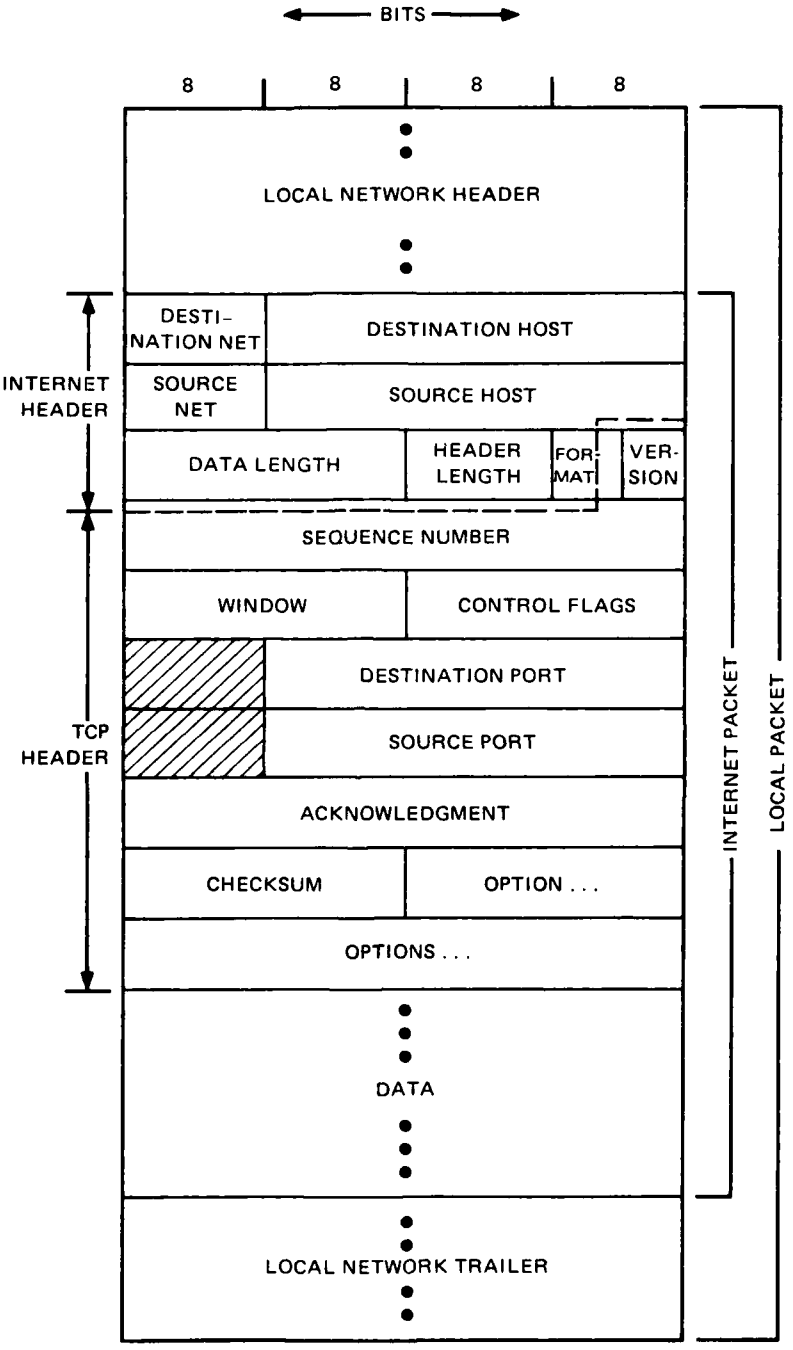


FIGURE 7 FORMAT OF INTERNET PACKET

When enough contiguous octets starting at the window base arrive to fill a user receive request, those octets are delivered to the user and their sequence numbers are acknowledged in an ACK to the originating TCP, implicitly advancing the window bounds within the sequence number space. So long as some of the octets within a packet remain unacknowledged, the originating TCP will continue to retransmit the packet after the appropriate time-out interval.

To achieve flow control, TCPs are prohibited from sending octets that are known to overflow the receiver's window. Therefore, a receiver automatically regulates the transmission rate of the sender through the receive acknowledgments which advance the window bounds. TCPs may alter their window sizes at any time; there is a protocol mechanism for informing the remote TCP of such changes.

2. Connection Management

The TCP is responsible for establishing and disbanding the connections between process ports at the direction of the calling processes, and for automatically detecting and reporting error conditions (such as a half-open connection resulting from loss of contact with the remote TCP). To open a connection, the cooperating TCPs must exchange matching local/foreign socket name pairs and synchronize window bases by agreeing upon ISNs for the transmit and receive windows; a three-way handshake mechanism was evolved for these functions.

The three-way handshake is essentially a unidirectional attempt to establish a connection; the initiating process makes a specific request for a connection between two process ports, and the responding process usually accepts, having previously issued a listening open (foreign socket name unspecified). The TCP can also establish a connection when a simultaneous initiation occurs. The following sequence of events defines the three-way handshake:

(1) The initiating TCP dispatches a SYN (synchronize) packet to the destination TCP; it includes the local and foreign port names, and the ISN for the transmit window: <SEQ 100><SYN><LOCAL PORT A><FOREIGN PORT B> (since all packets contain the local and foreign port names, they will not be shown in the rest of the examples).

(2) When the responding TCP receives the SYN, it checks whether a local process has issued a listening open with which to accept the foreign request for connection. If not, a RST (reset) packet is returned that causes the initiating TCP to abort the connection attempt. Otherwise, a combined SYN-ACK is returned that

acknowledges the foreign transmit window ISN and specifies the local transmit window ISN: <SEQ 300><SYN><ACK 101>. Note that the acknowledgment to sequence number "X" is "ACK X+1".

(3) The initiating TCP replies to the SYN-ACK by acknowledging the ISN specified in the SYN portion of the packet: <SEQ 101><ACK 301>. The connection is now established.

Sometimes, a process will issue an open with the foreign port specified, and a matching SYN will come in after the SYN for the open has been sent; both sides are simultaneously trying to initiate the connection. Instead of replying with a SYN-ACK, the TCP simply sends an ACK, since a SYN has already been dispatched. The connection is now established at this end, and will be established at the other end in the same manner owing to symmetry.

Connection closing is a three-way handshake superficially identical to that used for establishing a connection, except that the FIN (finish) rather than SYN control function is used. A connection close exchange is not terminated until all outstanding data in both directions have been delivered and acknowledged.

For the connection management mechanism to be robust, it must be immune to the arrival of spurious (very old) control packets (replay), and it must be capable of detecting half-open connections that arise when one side of a connection crashes and comes back up with no memory of the connection. The TCP has mechanisms utilizing the RST (reset) control function for these purposes. A RST is dispatched whenever a TCP receives confusing control packets; it immediately causes the connection or connection attempt to be aborted, and a reconnection attempt usually follows. Also provided is a RSN (resynchronize) control function used to resynchronize on new sequence numbers when a connection is not cycling through its sequence number space fast enough, a situation that can lead to packets with nonunique sequence numbers if a host crashes and reinitializes.

IV THE PRNET IN MILITARY APPLICATIONS

It is clear from the preceding sections that a PRNET data communication medium with appropriate protocols offers several advantages for military systems, including:

- Internetworking capability
- High sharing capacity
- High bandwidth
- High ETE communication reliability
- Relative immunity to communication equipment (network node) failure
- Capability for mobile terminals
- Automatic operation
- Resistance to channel jamming, spoofing, and detection.

More specifically, when applied to TACFIRE, the PRNET technology will permit the placement of all FDCs [14], VFMEDs [15], BDUs [16], and DMDs [17] on a single PRNET, endowing them all with the ability to intercommunicate at will, a capability missing in the current multinetwork TACFIRE configuration. This section outlines a proposed approach for replacing the VHF FM equipment with a PRNET.

A. Attachment to TACFIRE

The packet radio equipment must interface to the TACFIRE system at two levels: at the FDC AN/GYK-12 [18] computer, and at the various remote terminals (VFMED, BDU, and DMD). The primary concern is to achieve this with minimal effect upon the TACFIRE hardware and software components. Hardware effects can be avoided if all interfacing occurs at the component bus level, while software effects can be minimized if the hardware bus interfaces are made at points where complete TACFIRE messages are available for input and output.

The proposed TACFIRE configuration is shown in Figure 8. At the FDC level, a single PRNET HIU (similar to a TIU) will replace DDTs [19] attached to the AN/GYK-12 computer. This will require special, high-speed interfaces; SRI is currently devising an architecture for such a device in a related effort. At the remote terminal level, two separate interfacing strategies are necessary, one for the DMD, and another for the RDT of the BDU and VFMED:

(1) The DMD has an internal bus directly accessible through an accessory slot, or via an optional RS-232C interface. TACFIRE messages output to this bus from the DMD processor are in Hamming-coded, time-dispersed format, destined for the DMD modem card. Messages on the bus originating from the modem card are in ASCII character format, destined for the processor. Therefore, a HIU can be interfaced to this bus, inputting to it the received TACFIRE messages with no modification, and extracting from it a bit stream that can be uncoded back into an ASCII TACFIRE message.

(2) The interface at the BDU and VFMED must be made to the RDT. No specific information about the RDT's internal organization is available at this time; if the RDT is similar to the DMD, an analogous approach may be possible. Otherwise, it may be possible to replace the RDT completely by incorporating its message processing functions into the HIU (Section V-B).

The proposed configuration does not show two important elements of the present TACFIRE system: the security (COMSEC) and voice equipment. For the purposes of a testbed system, it may be necessary for all PRNET traffic to be unencrypted (clear), and voice radios separate from the RDT will have to be provided (if the RDT is indeed replaced by a HIU).* Since introduction of the PRNET equipment separates the voice and data channels, some of the TACFIRE communication components (specifically the CCU [20] and RCMU [21]) will not be used.

Packet radio technology is evolving specifically to meet military needs. Currently, there are projects for designing encryption equipment for packet-switching systems, and when available, it will be possible to integrate this hardware into the proposed TACFIRE configuration,

---

* If modifications to the AN/GYK-12 software are practical, it would be possible for the PRNET to distribute encrypted TACFIRE messages if at least the destination character of the message header were delivered to the HIU in the clear. Otherwise, this could only be achieved if a broadcast protocol were developed for the PRNET.

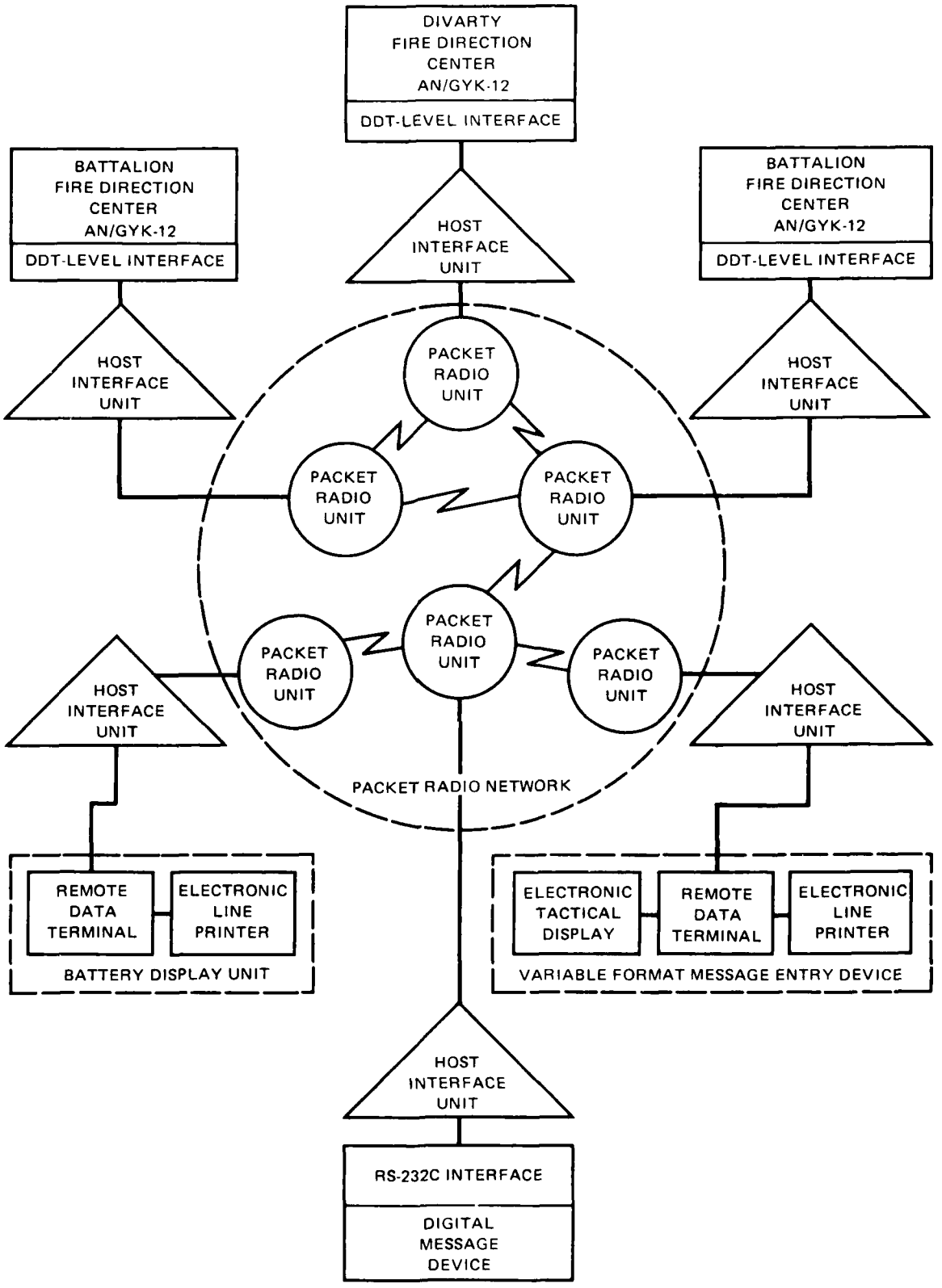


FIGURE 8 PROPOSED TACFIRE COMMUNICATION STRUCTURE

displacing the COMSEC equipment. There is also some possibility that future packet radio technology will have a limited digitized voice capability; if so, it might at that time be possible for the PRNET equipment to fulfill TACFIRE's voice needs. SRI is currently working on PRNET packetized speech research, but the efficacy of simultaneous speech and data traffic on the PRNET is still unclear.

B. Army PRNET Communication Protocols

Traffic in military communication networks, including TACFIRE, has several properties not found in current packet-switching applications. These include:

- Hierarchical communication structure: The bulk of traffic on a military network is radial, between a small number of command centers, and a large number of subscribers.

- Dispatch and query/response traffic flow: Communication with a given remote subscriber is usually unidirectional (in the form of dispatches). The largest amount of interaction is normally restricted to a message (query) in one direction evoking a response in the other direction (i.e., half-duplex operation).

- Demand traffic addressing: Traffic between two specific sites is usually sent as the need arises; two sites do not know in advance when or if they will have occasion to exchange traffic.

In the most common communication scenario, there is a fairly consistent traffic flow between command centers, and a heavy flow of traffic between each command center and the set of remote subscribers. However, traffic is exchanged between a command center and a specific remote subscriber only occasionally (no more than several times per minute), and two remote subscribers directly communicate even more rarely.

To be generally useful in military applications, the protocol structure of the military PRNET must be capable of handling these diverse traffic patterns reliably and efficiently. The proposed protocol structure, shown in Figure 9, is designed to be applicable to a wide variety of military communication needs, without the need for higher-level protocols.

Of course, TACFIRE employs a VHF-FM radio communication protocol integral to the system's operation. However, to avoid modification to TACFIRE software, all messages generated by the TACFIRE components, including protocol messages (ACKS and retransmissions), will be obtained

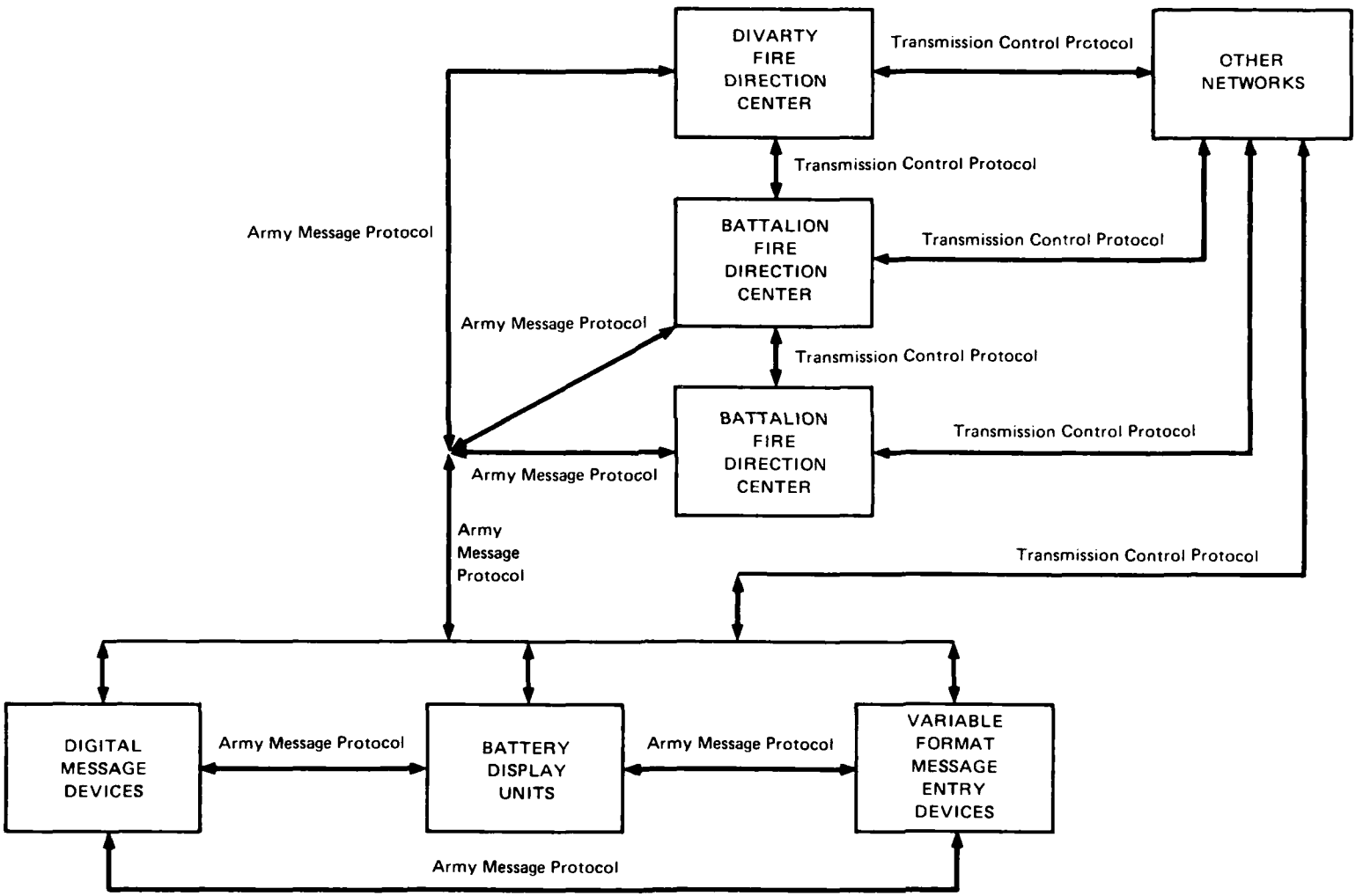


FIGURE 9 PROPOSED TACFIRE PROTOCOL STRUCTURE

by the HIUs at the interfaces described in Section V-A; they will be routed to their destinations without modification.

TCP will be used to handle all communication between command centers (the FDCs in the TACFIRE system), since intercenter traffic is persistent and regular; the use of TCP will also provide the desired internetting capability. A new Army message protocol (AMP, Section VI-A) will mediate all low-level communication between the command centers and remote subscribers (BDUs, VFMEDs, and DMDs), as well as any that is needed between the remote subscribers themselves (a remote subscriber requiring an internet capability could be provided with a TCP, or an internet forwarder could be installed at one of the FDCs).

The process structure within the HIU is shown in Figure 10. Interposed between the network interface and the TCP and AMP processes within the HIU will be a packet multiplexer (PKTMUX). This process will receive packets from the network and dispatch them individually to either the TCP or AMP, depending upon the destination ID in the PRNET header; for this purpose, a destination naming convention will be required (such as the one presented in Figure 11). Similarly, a message multiplexer process (MSGMUX) will reside between the TACFIRE component interface and the TCP and AMP processes. MSGMUX will have a table that will map destination IDs supplied in TACFIRE message headers [22] into the corresponding PRNET IDs; MSGMUX will be able to appropriately multiplex messages to the TCP or AMP based on the PRNET ID found in the table. The contents of the destination table will be set at system generation time, but will be modifiable at run time.*

MSGMUX will communicate with the attached TACFIRE component through an I/O driver that will be unique to each component. MSGMUX will exchange TACFIRE messages in ASCII format with this driver, which will perform any format transformations necessary for communication with the TACFIRE component. For example, the driver for the DMD may have to un-time-disperse and un-Hamming-code outbound messages before passing them to MSGMUX, but will be able to forward inbound messages directly from MSGMUX to the DMD. For the RDT, the driver may have to implement the entire RDT application, if the RDT is replaced by the HIU (Section V-A).

Owing to connection state information storage requirements of its current LSI-11 TCP implementation [23], the HIU can currently support only about five open TCP connections. To enhance this number, the Army TCP implementation (to be developed) will have a two-state connection activity mechanism. Connections for which enough outstanding traffic is enqueued (inbound or outbound) will be "active," and the Army TCP will

---

* MSGMUX may also have to simulate half-duplex communication network operation. TACFIRE's communication software is currently highly dependent on the half-duplex nature of the VHF-FM radio network; its ability to function properly with a full-duplex medium is uncertain, and requires further investigation.

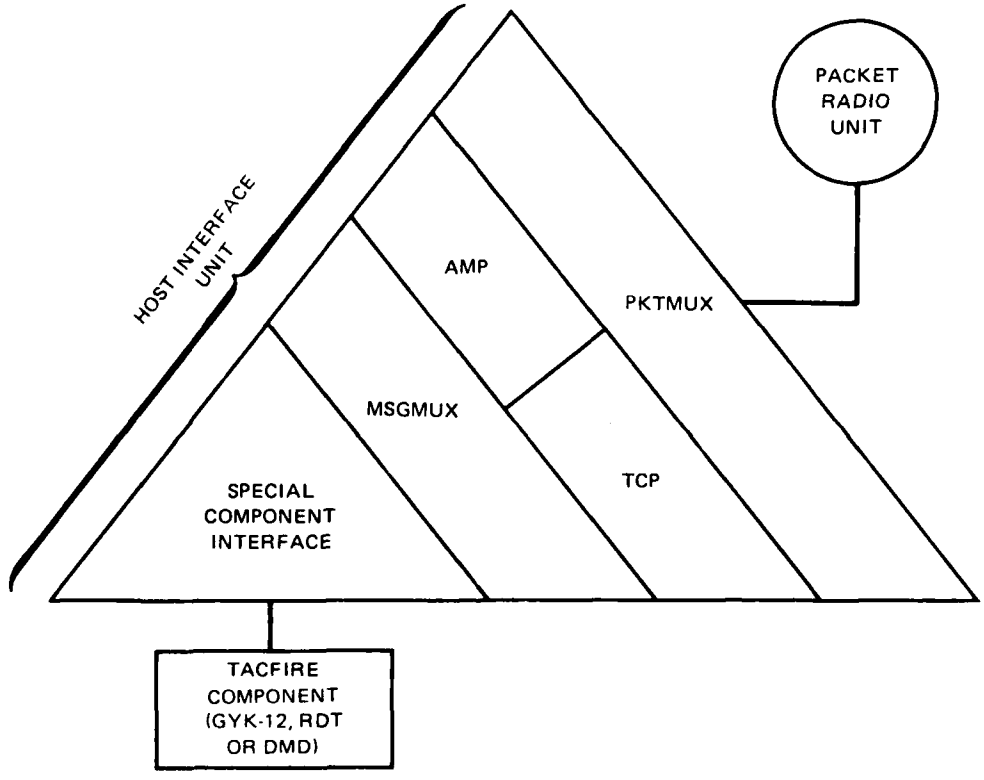


FIGURE 10 HOST INTERFACE UNIT PROCESS STRUCTURE

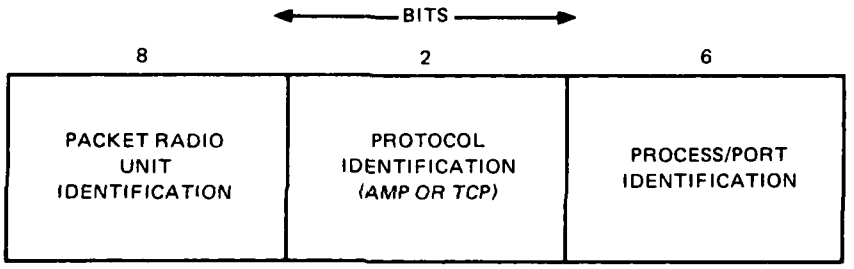


FIGURE 11 PROPOSED INTRANET NAMING CONVENTION

maintain full state information for them. Connections with little or no enqueued traffic will be "dormant," and only reduced state information will be stored. Connections will be assigned to the active state on a first-come-first-served basis until the Army TCP runs low on resources; at that point, connections will dynamically alter state depending upon their traffic load, until the Army TCP can again support all outstanding connections on a full-state basis. Using this mechanism, the number of connections supportable by the PRNET Army TCP implementation should increase more than tenfold (up to about 40 or 50 connections).

V THE ARMY MESSAGE PROTOCOL

AMP was designed for the remote subscriber communication application for four reasons:

- Even with the reduced-state TCP implementation outlined above, the HIU cannot support enough TCP connections to serve a large set of remote subscribers (i.e., tens to hundreds).
- The overhead required to open and close a TCP connection just to send a small amount of traffic is prohibitive.
- Owing to the nature of subscriber message traffic, there is no need for the sophisticated data pipelining and flow control mechanisms provided by the TCP.
- The operational characteristics of military subscriber traffic require protocol performance characteristics not found in any other existing protocols.

AMP is purely an intranet protocol and, like SPP, its operation depends upon implementation within a network having certain operational characteristics (Section VI-C). AMP is a connection-free PAR protocol that uses the PRNET header for addressing and sequencing, and an extra 16-bit AMP header for control. It is inherently simplex, and achieves flow control while avoiding reordering difficulties by prohibiting packet pipelining. Like the TCP, AMP accepts arbitrarily sized letters from processes and chops them into segments as necessary for transmission within the PRNET.

A. AMP Specification

The act of sending a letter between two processes is called a transaction. Letters are exchanged as transactions between two processes which have an association (as opposed to a connection). An association is created when the first packet of the first transaction arrives at a destination and is acknowledged; it persists until inactivity and a need for resources dictate the deletion of its state information at either end. An association is identified at the source AMP by its PRNET source and destination IDs, plus the sequence number of the last packet sent. At the destination AMP, it is identified by its

source and destination IDs, plus the sequence number of the last packet received. Associations are polarized (one-way), so letters enroute between two processes in opposite directions are independent transactions of different associations. Each association may have only one outstanding (unacknowledged) packet in the network at a time.

AMP uses the PRNET header source, destination, packet length, and sequence number fields in the standard PRNET manner: the source and destination fields identify unique process ports, and sequence numbers for successive packets of an association must monotonically increase by one. The AMP header (Figure 12) is the first 16-bit word of the PRNET packet, and contains the following information:

(1) The most significant bit (bit 15) is the ACK (acknowledgment) bit. When set, it indicates that this packet is an acknowledgment for the last packet received for the association denoted by the ACK packet's source and destination IDs; the PRNET header sequence number field is set equal to the sequence number of the acknowledged packet. ACK packets contain no text (other than the AMP header), and their exchange is considered to be association-free.

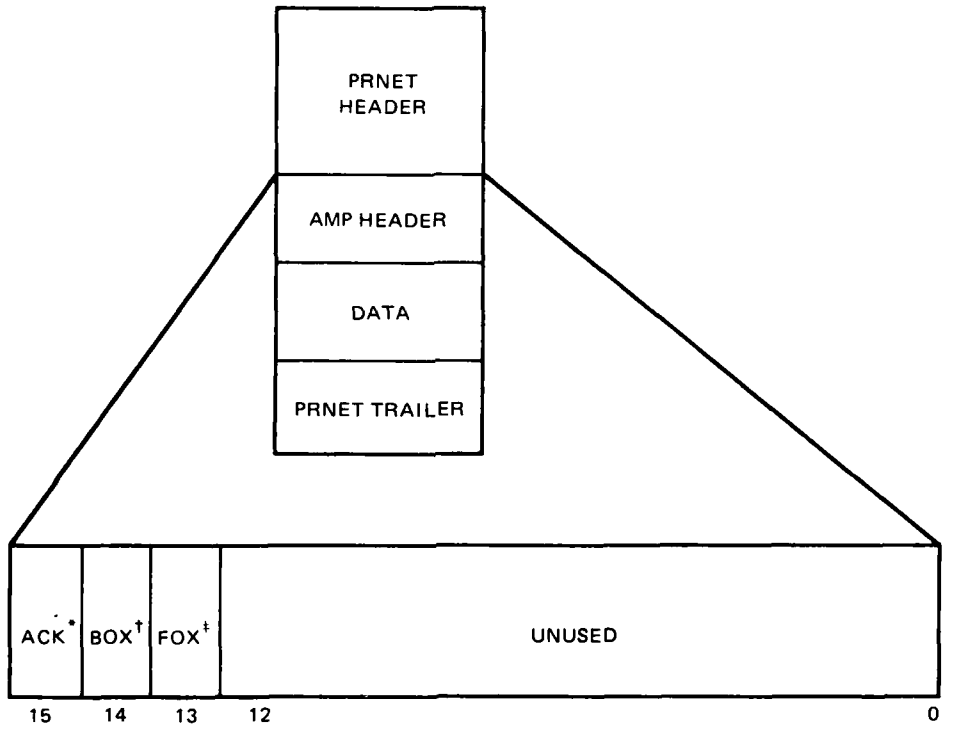
(2) Bit 14 is the BOX (beginning of transaction) bit. When set, it indicates that the packet contains the first segment of a new transaction.

(3) Bit 13 is the FOX (finish of transaction) bit. When set, it indicates that the packet contains the last segment of the current transaction.

For control purposes, AMP recognizes four time-out intervals:

(1) ITER (termination interval): When an association has fallen idle for a time greater than ITER (no traffic), the association may be terminated at either end by deletion of its state information.

(2) IRES (resynchronization interval): When an association with an outstanding transaction has fallen idle for a time greater than IRES, the destination AMP resynchronizes on the next new transaction (BOX packet) received for that association, regardless of its sequence number, aborting the outstanding transaction and flushing its data. The expiration of IRES is cancelled at the destination AMP if the next packet for the outstanding transaction finally arrives.



* ACK = Acknowledgment

† BOX = Beginning of transaction

‡ FOX = Finish of transaction

FIGURE 12 FORMAT OF AMP HEADER

(3) IREX (retransmission interval): The AMP at a transaction source will retransmit the last packet that was sent for a given transaction if no acknowledgment for the packet has arrived and a time IREX has elapsed since it was sent.

(4) IPER (persistence interval): If no acknowledgment for a packet or its retransmissions has been received, and an interval IPER has elapsed since the first sending attempt, the outstanding transaction is aborted. IPER is equal to the product of IREX and the maximum allowed number of retransmission attempts (NREX).

For each association, AMP maintains:

. a timer for determining the expiration of ITER

. a timer for determining the expiration of IREX

. an association state block (ASB, Figure 13) containing:

- the PRNET ID of the transaction source
- the PRNET ID of the transaction destination
- the sequence number of the last packet sent or received (depending on whether the AMP is on the source or destination end of the association)
- a count of how many times the last packet sent for the association has been retransmitted (need be kept at source end only)
- a flag (XSUS) that indicates that the outstanding transaction has been suspended owing to expiration of the resynchronization timer (need be kept at destination end only)
- a flag (ASUS) that indicates that the association has been suspended owing to expiration of the termination timer
- a pointer to the reassembly buffer for the outstanding transaction

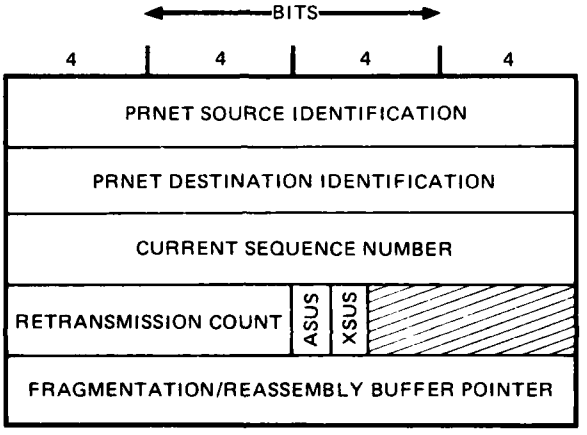


FIGURE 13 FORMAT OF ASSOCIATION STATE BLOCK

The AMP at a letter source will initiate a transaction on request between two processes if there is currently no outstanding transaction between them (otherwise, the request is enqueued). To do so, the first segment of the letter is packetized, the BOX bit of the AMP header is set, and an ISN is chosen. If an association already exists between the processes, the ISN is simply the current sequence number for the association (in the ASB) plus one; otherwise, a new association is created by allocation of an ASB for its state information, and an arbitrary ISN is chosen. As soon as the packet is sent, a retransmission timer and a termination timer for the association are set.

When a destination AMP receives a BOX packet for a new association, it allocates an ASB, filling in the sequence number from the BOX packet, and buffers the first segment of the transaction for reassembly. An ACK is returned to the transaction source, and a resynchronization timer and termination timer for the association are set.

Destination AMPs will only accept and reassemble a packet for an association if either:

(1) it arrives in proper sequence: A packet is in proper sequence if its sequence number is one greater than the stored sequence number for the association. When a packet arrives in sequence, its segment is added to the reassembly buffer for the association, an ACK for it is returned, and the resynchronization and and termination timers are reset.

(2) it is a BOX packet whose sequence number is greater than the stored sequence number for the association (this could happen if the source AMP aborts the outstanding transaction, or else crashes and recovers quickly). The outstanding transaction is aborted by flushing its data stream, and the destination AMP resynchronizes upon the new transaction by storing the BOX packet's sequence number and data segment.

Packets that arrive with sequence numbers equal to the stored number are duplicates; they are acknowledged and discarded. Packets with sequence numbers less than the stored number are very old duplicates or garbage, and are simply discarded.

When the resynchronization timer for an association expires, its outstanding transaction (if any) is suspended (by setting its XSUS flag); the destination AMP will resynchronize on the next BOX packet for that association, regardless of sequence number, by aborting the

suspended transaction and storing the ISN and first data segment of the new transaction. However, if the next packet in sequence for a suspended transaction arrives before a new BOX packet, the resynchronize time-out is cancelled (the XSUS flag is cleared), and the transaction is resumed normally.

For each association, the source AMP may send only one packet to the network at a time, and must await its acknowledgment before sending another; successive packets for an association must bear sequence numbers that monotonically increase by one. If an ACK for a packet does not arrive within IREX seconds of its transmission, the packet is retransmitted, and the retransmission timer is reset. When an ACK is finally received, the next packet in sequence is sent, and the retransmission and termination timers are reset; however, if there are no more packets to send, only the termination timer is reset, and the retransmission timer is cancelled. A transaction is aborted if no ACK has been received despite retransmissions after an interval IPER (i.e., after NREX retransmissions every IREX seconds); the calling process is notified of the failure, and the next transaction for the association (if any) is initiated.

If the termination timer for an association ever expires at either end, the association is suspended (by setting its ASUS flag). Whenever an AMP requires storage resources, suspended associations are terminated by deletion of their ASB and flushing of their data streams, and the memory space is reassigned as necessary.

An association can become suspended at a source AMP if (1) there is a long time gap between successive transactions, or if (2) a long time has passed since an ACK has been successfully received from the destination AMP, despite repeated transmission attempts spanning successive transactions. It will be resumed if, before being terminated, (1) a new transaction is initiated by the user, or if (2) an ACK finally arrives. Transaction exchange proceeds normally on a resumed transaction, and it is no longer susceptible to termination (i.e., the ASUS flag is cleared).

At a destination AMP, an association may become suspended if a long time has elapsed since receipt of a valid packet. The association will be resumed if a new transaction (BOX packet) arrives to resynchronize upon, or if the next packet in sequence for the outstanding transaction (if any) is finally received.

When the source AMP sends the last segment of a letter, it sets the packet's FOX bit (if the entire transaction fits within one packet, both the BOX and FOX bits within that packet's AMP header will be set). When a properly sequenced FOX packet arrives at a destination AMP, it sends an ACK back as usual, reassembles the last segment, and delivers the letter to the user process. The transaction (but not the association) is now terminated at the destination; the destination AMP awaits receipt

of the next packet in sequence, which should be a BOX packet for a new transaction (only a BOX packet will be accepted). When the FOX packet ACK is received by the source AMP, the transaction (but not the association) is terminated at the source; the source AMP initiates the next enqueued transaction, or awaits another user transaction request. Of course, ASB information at either end is always updated whenever a new transaction is initiated.

B. AMP Process Interface

AMP provides four user calls:

(1) SENDX--Send Transaction: Supplied with the call are:

(a) the source port name

(b) the destination ID (includes network address and port name)

(c) the letter length (octets)

(d) a pointer to the letter

SENDX returns status that indicates whether or not the letter was successfully delivered. The user may try again if the delivery failed.

(2) RECVX--Receive Transaction: A destination AMP will not set up associations for a local port unless the port has been defined with a RECVX. Once defined, all transactions bound for that port will be accepted by establishing associations. RECVX takes the destination port name as a parameter, and delivers each received letter by returning a source ID, octet count, and letter pointer.

(3) ABORX--Abort Send Transaction: The transaction sent on the specified local port is aborted by flushing its data stream; nothing is returned. The AMP will begin the next transaction (if any).

(4) STATA--Status of Association: Supplies the status of the specified association by returning a pointer to a copy of the ASB for the association.

C. Analysis of AMP

AMP has three parameters whose values must be chosen:

- IRES, the resynchronization interval
- IPER, the retransmit persistence interval
- IREX, the retransmit interval.

Their optimum values, as well as AMPs performance, are a function of the distributions of the following network operational parameters:

- Network packet lifetime (PKTLIF)
- Packet round trip delay (PKTDEL)
- Soft outage (connectivity loss) time (SOTIM)
- Hard outage (HIU crash/recovery) time (HOTIM).

For AMP to function properly, the subnet must have two important operational characteristics. The first is that the maximum packet lifetime must be less than the minimum hard outage time,

$$\text{maximum PKTLIF} < \text{minimum HOTIM} .$$

Otherwise, a destination AMP could crash and recover, and get confused by old packets still in the network. As long as this is impossible, AMP can handle all other resynchronization problems through its IRES time-out. To avoid accidental resynchronization on a duplicate packet, IRES must be large enough to allow the network to clear of all duplicates:

$$\text{IRES} > \text{IPER} + \text{maximum PKTLIF} .$$

However, to ensure that the destination AMP will be ready to resynchronize as soon as a source has recovered from a crash, there is an upper bound on IRES:

$$\text{IRES} < \text{minimum HOTIM} + \text{maximum PKTLIF} .$$

IRES should be chosen between these two bounds by experiment.

The second subnet operational requirement is sufficient separation between the spectrums of hard and soft outage times. To oversimplify,

maximum expected SOTIM < minimum expected HOTIM .

If this condition is not met, it is impossible to set IPER greater than the maximum SOTIM (see below) while making IRES lie between IPER and the minimum HOTIM; thus, the desired reliability would not be achieved.

Whether the PRNET can meet this requirement depends upon the definition of where the borderline between hard and soft outages lies: most connectivity losses are on the order of seconds long, whereas the HIU hardware crash/recovery cycles last several minutes. However, it is not impossible for a mobile terminal to park under a bridge and lose connectivity for a long time. For the purposes of AMP, this would have to be categorized as a hard outage, and transactions outstanding during the outage time would be lost and not delivered (unless the application process saved them and tried again later). Except for such unusual occurrences, the PRNET fulfills the above requirement.

AMP's maximum throughput is almost purely a function of PKTDEL, since the next packet in sequence cannot be sent until an ACK for the one preceding it has been received.* Because PKTDEL in the PRNET is quite small, AMP should easily handle the transactional traffic for which it was designed (for applications requiring very high bandwidth, or for those within networks having a high PKTDEL, a pipelining protocol such as TCP would be more appropriate).**

Throughput degrades as retransmission becomes necessary, both to make up for packet loss by the subnet and to cover soft outages. A lower bound on the retransmission persistence, based solely on soft outages, may be expressed as

IPER > maximum expected SOTIM .

This actually defines the minimum time span over which packets must successfully traverse all but the last hop of the subnet route; IPER at the source should be longer to account for packet loss along the route, and its upper bound is best picked experimentally to achieve the required reliability. To avoid network loading by excessive

---

* For a rigorous analysis of PAR protocol throughput as a function of round trip delay and loss characteristics, see [13], pp. 86-91.

** This is why TCP is necessary for internet communication; gateways may add significantly to PKTDEL. Additionally, PKTLIF can become very large. In some internet environments, however, especially packet radio multinetworks, these problems may not arise, and AMP might be applicable.

retransmissions, IPER should not be set higher than necessary.*

IREX must be chosen to avoid generating a new retransmission before the ACK for the previous one can normally arrive. The lower bound is therefore

IREX > average PKTDEL .

IREX should be set close to the lower bound, since a few duplicate packets do not hurt too much, but waiting unnecessarily long between retransmissions can greatly slow down a nonpipelined protocol.

---

* Of course, the lower bound on IPER could be at a point where loading is already unacceptable. However, since other PAR protocols (such as TCP and SPP) work well within the PRNET, this problem will probably not arise.

VI SUMMARY

The advantages of Packet Radio Network (PRNET) communication in military applications were discussed. A new Army Message Protocol (AMP) for the PRNET was specified and informally analyzed. This packet transport protocol was designed specifically to serve the needs of transaction-oriented military tactical communication on PRNETs. A transparent approach for attaching a PRNET to the TACFIRE system at logical bus interface levels was presented; the PRNET protocol structure suggested for this application uses AMP to mediate TACFIRE's intranet transactions with remote terminals, and the Transmission Control Protocol (TCP) for all inter-FDC and internetwork communication. This approach is fairly general, and applicable to other military systems (e.g., TOS). TACFIRE was chosen as a test vehicle for military packet radio communication since it is a well established system, with much available documentation. The recommendations of this report were of a preliminary nature; they were intended as starting points for more detailed design and analysis in the light of more complete information and initial tests.

REFERENCES

1. "The Tactical Fire Direction System (TACFIRE), Reference Note," U. S. Army Field Artillery School, Gunnery Dept., Fort Sill, Oklahoma (May 1972).

2. R. Kahn, "The Organization of Computer Resources into a Packet Radio Network," IEEE Transactions on Communications, Vol. COM-25, No. 1 (January 1977).

3. R. Kahn, "Resource Sharing Computer Communication Networks," Proc. IEEE, pp. 1397-1407 (November 1972).

4. F. E. Heart et al., "The Interface Message Processor for the ARPA Computer Network," 1972 Spring Joint Computer Conference, AFIPS Conference Proceedings, vol. 40 (1972).

5. "Specifications for the Interconnection of a Host and an IMP", Report No. 1822, Bolt, Beranek, and Newman, Inc. (April 1972).

6. J. Burchfiel, R. Tomlinson, and M. Beeler, "Functions and Structure of a Packet Radio Station," in AFIPS Conference Proceedings, vol. 44, AFIPS Press (1975).

7. R. Sunlin, "Packet Radio Protocol Program, PRCAP3," Collins Radio Group Government Telecommunications Division (February 1977).

8. R. Binder et al., "Aloha Packet Broadcasting--A Retrospect," in AFIPS Conference Proceedings, vol. 44, AFIPS Press (1975).

9. L. Kleinrock and F. Tobagi, "Carrier Sense Multiple Access for Packet Radio Channels," in Proc. Int. Conf. Commun., Minneapolis, Minnesota (June 1974).

10. M. Beeler, "SPP Definition," Packet Radio Temporary Note No. 177 (April 1976).

11. V. Cerf, "Specification of Internet Transmission Control Program, TCP (Version 2)" (March 1977).

12. V. Cerf and R. Kahn, "A Protocol for Packet Network Intercommunication," IEEE Trans. Commun., vol. COM-22, pp. 637-648 (May 1974).

13. C. Sunshine, "Interprocess Communication Protocols for Computer Networks," Technical Report No. 105, Digital Systems Laboratory, Stanford, California, December 1975 (Ph. D. Thesis).

14. "Subsystem Specification for the Bn FDC Subsystem for the Fire Direction System, Artillery AN/GSG-10(V)," Data Systems Division, Litton Systems (July 1969).

15. "CEI Specification for the Variable Format Message Entry Device (VFMED)," Spec. No. EL-CP-00041107C, Doc. No. 586011-600C, Data Systems Division, Litton Systems (August 1976).

16. "CEI Specification for Battery Display Unit," Spec. No. EL-CP-00041202B, Doc. No. 586014-600B, Data Systems Division, Litton Systems (July 1969).

17. "Prime Item Development Specification for the Digital Message Device (DMD), for the Tactical Fire Direction System (TACFIRE)," Spec. No. EL-SS-2603-TF (April 1975).

18. "CEI Specification for Computer," Spec. No. EL-CP-00041101B, Doc. No. 586017-600B, Data Systems Division, Litton Systems (February 1971).

19. "CEI Specification for Data Terminal Unit (DTU)," Spec. No. EL-CP-00041109B, Doc. No. 586068-600B, Data Systems Division, Litton Systems (November 1970).

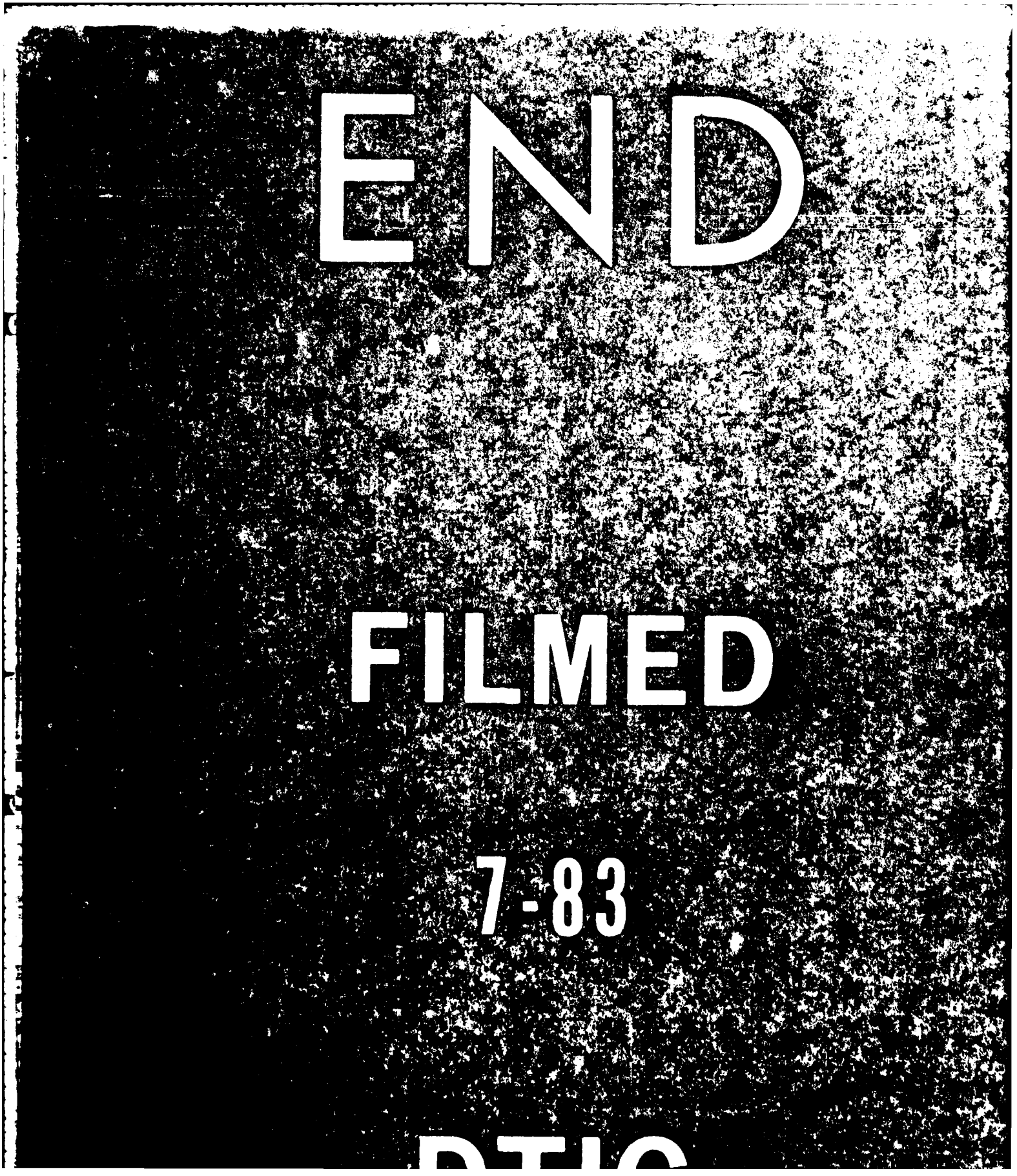
20. "CEI Specification for the Communications Control Unit (CCU)," Spec. No. EL-CP-00041129, Doc. No. 586438-600, Data Systems Division, Litton Systems (September 1976).

21. "CEI Specification for the Remote Communications Monitor Unit

(RCMU)," Spec. No. EL-CP-00041128, Doc. No. 586318-600, Data Systems Division, Litton Systems (July 1976).

22. "General Specification for Message and Code Standards," Spec. No. EL-SS-004-R-3B, Doc. No. 586000-603B, Data Systems Division, Litton Systems (March 1970).

23. J. Mathis, "Single-Connection TCP Specification," Stanford Research Institute, Menlo Park, California (July 1977).


END
FILMED
7-83
DTIC